<u>AMENDMENTS TO THE CLAIMS</u>

1.   (Currently amended) A computer-implemented method of collecting and storing information about the programs installed on and the services provided by a computer for subsequent retrieval, comprising:

     (a)   extracting from the computer information including, but not limited to, information about the computer operating system, hardware, and processor and storing the system information in a log file in a standardized language;

     (b)   extracting from the computer executables information including, but not limited to, information about executables included in a defined set of folders stored on the computer and executables associated with services provided by the computer<u>;</u>

     <u>(c)</u>   <u>for each executable whose information is extracted, determining if the executable is associated with the operating system,</u> and <u>if it is determined that the executable is not associated with the operating system,</u> storing ~~the executables~~ information <u>about the executable</u> in the log file, the ~~executables~~ <u>stored executable</u> information including attributes determined by the executables, each executable including a plurality of associated attributes, the attributes including at least one attribute, other than a version number, in the standardized language;

     [[(c)]] <u>(d)</u>   extracting from the computer information regarding the application programs installed on the computer including linked executables and storing the application program information in the log file in a standardized language, the application program information including attributes determined by the application programs including the linked executables; and

[[(d)]] (e)   deriving a signature for each of the executables, the signature being based on more than one of the plurality of attributes associated with the related executable and storing the resultant signatures in the log file in the standardized language.

2.   (Previously presented) The method of claim 1, further comprising storing the signatures in an XML file associated with the executables.

3.   (Previously presented) The method claim 1, further comprising storing the information in an XML file associated with the executables.

4.   (Previously presented) The method of claim 3, further comprising storing the signature in the XML file.

5.   (Previously presented) The method of claim 1, wherein extracting the application program information comprises accessing an installer component of the computer.

6.   (Previously presented) The method of claim 5, wherein the application program information is stored in connection with the installer component.

7.   (Previously presented) The method of claim 1, wherein extracting the application program information comprises accessing more than one information source.

8.   (Previously presented) The method of claim 7, further comprising choosing a best source of the more than one information source and utilizing that best source to provide at least some of the application program information.

9.   (Previously presented) The method of claim 8, further comprising storing information about the sources other than the best source with the application program information.

10.   (Previously presented) The method of claim 1, wherein deriving a signature comprises generating a number from the subset utilizing a cyclic redundancy check.

11-20. (Canceled)

21.   (Currently amended) A computer-implemented method of collecting and storing information about the applications installed on and the services provided by a computer for subsequent retrieval, comprising:

(a)   extracting from the computer executables information;

(b)   for each executable, determining if the executable is associated with a computer operating system; and

(c)   for each executable not associated with the operating system:

(i)   enumerating ~~executables~~ each executable associated with each application of a plurality of applications installed on the computer and each service of a plurality of services provided by the computer that has an associated executable;

(ii)   ~~for each executable,~~ extracting information about the executable, the information including a plurality of attributes associated with the executable, the attributes including at least one attribute other than a version number, and storing the information in a log file in a standardized language; and

(iii)   deriving a signature ~~for each executable~~ from a combined set of attributes, the combined set of attributes including at least two of the attributes associated with the related executable and storing the signature in the log file in a standardized language.

22.   (Previously presented) The method of claim 21, wherein the log file in which the signature is stored is an XML file associated with the application or service.

23.   (Previously presented) The method of claim 21, wherein the log file in which the information is stored is an XML file associated with the application or service.

24.   (Previously presented) The method of claim 23, further comprising storing the signature in the XML file.

25.   (Previously presented) The method of claim 21, wherein extracting the information that is stored in the log file comprises accessing an installer component of the computer.

26.   (Previously presented) The method of claim 25, wherein the information is stored in connection with the installer component.

27.   (Previously presented) The method of claim 21, wherein extracting the information that is stored in the log file comprises accessing more than one source for the information.

28.   (Previously presented) The method of claim 27, further comprising choosing a best source of the more than one source for the information, and utilizing that best source to provide at least some of the information that is stored in the log file.

29.   (Previously presented) The method of claim 28, further comprising storing information in the log file about the sources other than the best source with the information.

30.   (Previously presented) The method of claim 21, wherein deriving a signature that is stored in the log file comprises generating a number from the combined set utilizing a cyclic redundancy check.

31-39. (Canceled)

MSFT\25549AM6.DOC

40.   (New) A computer-readable medium having computer executable components for collecting information about one or more applications on a computer, comprising:

a selection component configured to select the most reliable source for obtaining information regarding attributes about an application on the computer and cause the collection component to retrieve the particular attributes from the most reliable source;

a collection component operable to:

extract from the computer information including the computer operating system, hardware, and processor and storing the system information in a log file in a standardized language;

for each executable on the computer, determining if the executable is associated with the operating system;

if the executable is not associated with the operating system, the collection component is further configured to generate a signature based on a set of attributes that are unique to the executable obtained from the source identified by the selection component;

wherein to generate a signature based on a set of attributes that are unique to the executable includes enumerating through a plurality of files to identify each executable that is linked to the application; and

a compatibility component configured to compare signatures of each executable obtained by the collection component to a compatibility database and determine whether an incompatibility exists between two or more applications installed on the computer based on the result of the signature comparison.

41.   (New) The computer-readable medium as recited in Claim 40, wherein to generate a signature based on a set of attributes that are unique to the executable from the source identified by the selection component includes providing a standard XML reporting schema to

represent attributes of an application in a hierarchical manner and wherein to determine whether an incompatibility exists between two or more applications installed on the computer includes serializing and deserializing information obtained by the collection component between XML and a non-hierarchical data format.